

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A system for parsing an arbitrary input stream, comprising:
 - a plurality of parsers operable to parse an input stream, each parser corresponding to a unique input structure;
 - a parser selection agent operable to receive the input stream and select a subset of the plurality of parsers to parse the input stream, wherein the input stream comprises a plurality of differing input structures and wherein the selected subset of parsers produce multiple parser outputs corresponding to the plurality of differing input structures; and
 - an encoding agent operable to convert the multiple parser outputs to a common grammar.
2. (Previously Presented) The system of Claim 1, wherein the multiple parser outputs correspond to differing grammars.
3. (Previously Presented) The system of Claim 1, wherein the parser selection agent and plurality of parsers are configured in a factory pattern and wherein the input stream comprises a plurality of messages having a plurality of headers comprising differing types of information.
4. (Previously Presented) The system of Claim 1, wherein the parser selection agent is operable to provide to a client, in response to a parse request, at least one of a parser output and an indication when at least some of the input stream is not successfully parsed and wherein the parser selection agent, prior to selection of the subset of parsers, is not informed in advance of the source or input structure associated with the at least some of the input stream.
5. (Previously Presented) The system of Claim 1, wherein the parser selection agent is operable to provide to a client, in response to a parse request, an error message when the parser selection agent is unable to identify one of the plurality of parsers to parse at least some of the input stream and when the parser selection agent is

able to identify more than one of the plurality of parsers to parse at least some of the input stream.

6. (Previously Presented) The system of Claim 1, wherein the input stream comprises fault information, the fault information being related to at least one of an alarm and an error, the fault information comprising first fault information related to a first event and in a first format and second fault information related to a second event discrete from the first event and in a second format different from the first format and wherein the encoding agent is operable to convert the first and second formats to a common format.

7. (Original) The encoding agent of Claim 6, wherein the first and second fault information uses different characters to refer to a same type of event and the encoding agent is further operable to convert the different characters to a common set of characters to refer to the event.

8. (Previously Presented) A method for parsing an arbitrary input stream, comprising:

- (a) receiving an input stream, the input stream comprising information defined by at least first and second input structures;

- (b) providing at least a portion of the input stream to each of a plurality of parsers, the plurality of parsers corresponding to differing sets of grammars;

- (c) receiving output from each of the plurality of parsers; and

- (d) based on the outputs of the plurality of parsers, performing at least one of:
 - (i) selecting a first output from a first parser that corresponds to the first input structure and a second output from a second parser that corresponds to the second input structure; and

- (ii) selecting a first parser corresponding to the first input structure to parse one or more first segments of the input stream and a second parser corresponding to the second input structure to parse one or more second segments of the input stream.

9. (Original) The method of Claim 8, wherein substep (d)(i) is performed.

10. (Original) The method of Claim 8, wherein substep (d)(ii) is performed.

11. (Previously Presented) The method of Claim 8, wherein the input stream comprises a plurality of nonstandardized headers.

12. (Previously Presented) The method of Claim 8, wherein the input stream is free of embedded tag indicating a source and/or input structure associated with the input stream and wherein step (b) comprises:

identifying one or more tokens in the input stream; and

based on the identified one or more tokens, selecting the at least one of a plurality of parsers.

13. (Previously Presented) The method of Claim 8, wherein step (b) comprises:

determining for each of the at least one of a plurality of parsers whether a match or a no match condition exists, a match condition indicating that a selected parser has successfully parsed a selected segment of the input stream and a no match condition indicating that the selected parser has not successfully parsed the selected segment of the input stream; and

applying the following rules:

when, for a parsed segment, only one match condition is found to exist, not generating an error message;

when, for a parsed segment, a match condition is not found to exist, generating an error message; and

when, for a parsed segment, multiple match conditions are found to exist, generating an error message.

14. (Original) The method of Claim 9, wherein a third parser successfully parses a first portion of the input stream to form a third output and the first parser successfully parses the first portion of the input stream to form a first output and further comprising:

determining which of the first and third outputs most likely corresponds to the first portion.

15. (Previously Presented) The method of Claim 14, wherein the determining step is performed using a least squares fit analysis and wherein step (d) is performed using a declarative programming rather than procedural programming approach.

16. (Original) The method of Claim 8, wherein the first parser produces a first output and the first output is a parse tree and further comprising:

recursively evaluating at least some of the nodes in the parse tree to identify nodes requiring additional parsing.

17. (Original) The method of Claim 8, wherein the first parser produces a first output and the first output is a parse tree and further comprising:

recursively examining at least some of the nodes in the parse tree to identify nodes of interest to a client.

18. (Original) The method of Claim 8, wherein the first parser produces a first output and the first output is a parse tree and wherein at least first and second nodes of the parse tree have differing formats and further comprising:

iteratively traversing a plurality of the nodes of the parse tree to locate nodes of interest, the nodes of interest comprising the first and second nodes; and

converting each of the located nodes of interest to a standard format.

19. (Original) The method of Claim 18, wherein each of the first and second nodes use different characters to refer to a same type of event and further comprising:

converting the characters in the first and second nodes to a common set of characters to refer to the type of event.

20. (Original) The method of Claim 8, wherein each of the plurality of parsers corresponds to a unique set of tokens and grammar rules.

21. (Original) The method of Claim 8, wherein each of the plurality of parsers corresponds to a unique set of attribute grammars.

22. (Previously Presented) A computer readable medium containing processor executable instructions to perform the steps of Claim 8.

23. (Previously Presented) A method for parsing computer generated information, comprising:

receiving a stream of information, the stream being generated by one of a plurality of possible different computational sources, wherein each computational source generates a stream corresponding to a unique input structure and wherein each of a plurality of differently structured segments of the stream is free of an embedded tag indicating a corresponding computational source and/or input structure for the respective segment;

comparing at least a portion of the stream with a set of tokens to provide a subset of tokens identified in the at least a portion of the stream;

heuristically identifying, from among at least one of a plurality of possible input structures and a plurality of possible computational sources, at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream; and

parsing the stream based on the identified at least one of an input structure and computational source.

24. (Previously Presented) The method of Claim 23, wherein the input stream comprises a plurality of headers, wherein the headers comprise differing types of information, wherein each of the tokens has a corresponding method expressing a set of syntactical and/or semantical relationships relating to the respective token and wherein the heuristically identifying step comprises:

for each token in the subset of tokens, invoking a corresponding method.

25. (Previously Presented) The method of Claim 24, wherein the comparing and heuristically identifying steps are performed using a declarative programming approach rather than a procedural programming approach, wherein the headers are nonstandardized, and wherein the invoking step comprises

setting, by an invoked method, a set of flags depending on the presence or absence of a syntactical and/or semantical relationship; and

wherein the values of the flags are used to heuristically identify the at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream.

26. (Currently Amended) The method of Claim 23, wherein a parser performing the steps of Claim 20 is not provided, ~~by another computational entity, with a flag external to the input stream to identify or assist in the identification of the at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream.~~

27. (Previously Presented) A computer readable medium containing processor executable instructions to perform the steps of Claim 23.

28. (Previously Presented) An autonomous heuristic parser, comprising:

an input operable to receive a stream of information, the stream being generated by one of a plurality of possible different computational sources, wherein each computational source generates a stream corresponding to a unique input structure; and

a parser operable to (a) compare at least a portion of the stream with a set of tokens to provide a subset of tokens identified in the at least a portion of the stream; (b) heuristically identify, from among at least one of a plurality of possible input structures and a plurality of possible computational sources, at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream; and (c) parse the stream based on the identified at least one of an input structure and computational source, wherein the parser is not provided with an input structure identifier, other than the corresponding input structure itself, either in or external to the at least a portion of the input stream to identify or assist in the identification of the at least one of the respective input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream.

29. (Original) The parser of Claim 28, wherein each of the tokens has a corresponding parser expressing a set of syntactical and/or semantical relationships relating to the respective token and wherein the parser is further operable, for each token in the subset of tokens, to (d) to invoke a corresponding method.

30. (Previously Presented) The parser of Claim 29, wherein the parser is further operable to (e) assign, by an invoked method, a set of flags with a corresponding set of values depending on the presence or absence of a syntactical and/or semantical relationship, wherein the values of the flags are used to heuristically identify the at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream.

31. (Previously Presented) The parser of Claim 28, wherein the parser is not provided, by another computational component, with a flag external to the input stream to identify or assist in the identification of the at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream.